

7.2. Reguli de dezvoltare software

7.2.1. Dezvoltarea aplicatiilor software – vedere generala

- **Buclo de control simpla** – software-ul are o singura bucla care cheama subrutine, fiecare subrutina manageriaza o parte din hard sau soft
- **Sistem controlat de intreruperi** – task-urile efectuate de catre sistem sunt chemate de diferite tipuri de evenimente (ex. Extern, timer, UART, etc.)
- **Multitasking**
 - Nonpreemptive - similar cu bucla de control simpla
 - Preemptive – are nevoie de un “programator” (eng. scheduler) pentru task-uri.
- **Microkernel**
 - Contine o simpla abstractizare a hardware-ului, avand un set primitiv the rutine ce implementeaza o parte minimala a uni OS (managementul memoriei, multitasking si comunicari interproces).
- **Kernele monolitice**
 - Toate serviciile OS ruleaza in kernel

7.2. Reguli de dezvoltare software

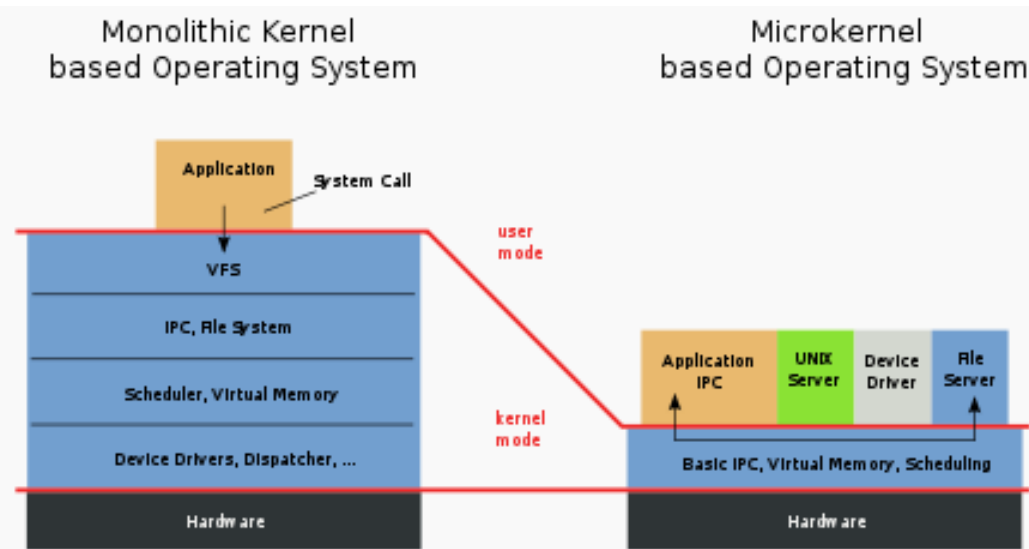
7.2.1. Dezvoltarea aplicatiilor software – vedere generala

- **Microkernel**

- Contine o simpla abstractizare a hardware-ului, avand un set primitiv the rutine ce implementeaza o parte minimala a uni OS (managementul memoriei, multitasking si comunicari interprocese).

- **Kernele monolitice**

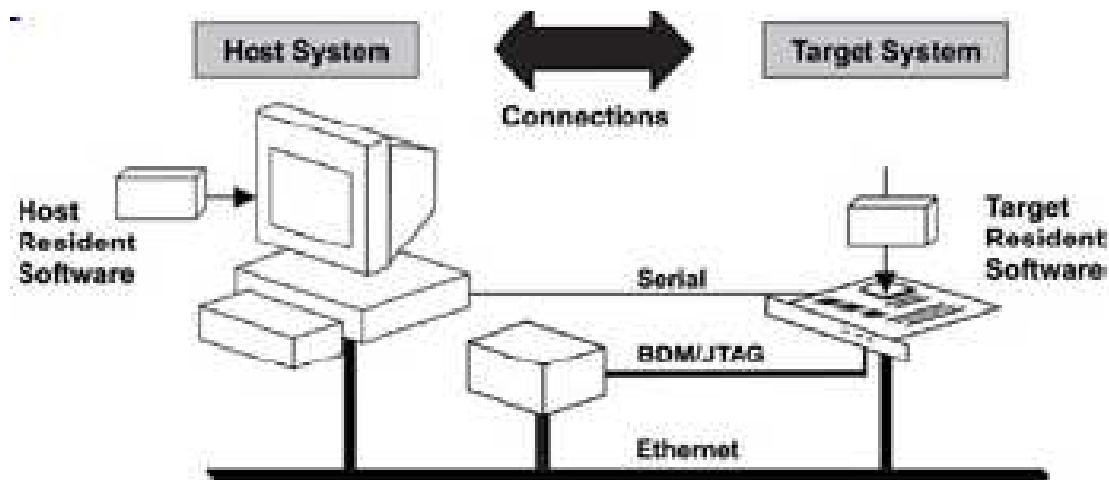
- Toate serviciile OS ruleaza in kernel



7.2. Reguli de dezvoltare software

7.2.1. Dezvoltarea aplicatiilor software – vedere generala

Dezvoltarea software se face folosind metodologia **cross-platform** :



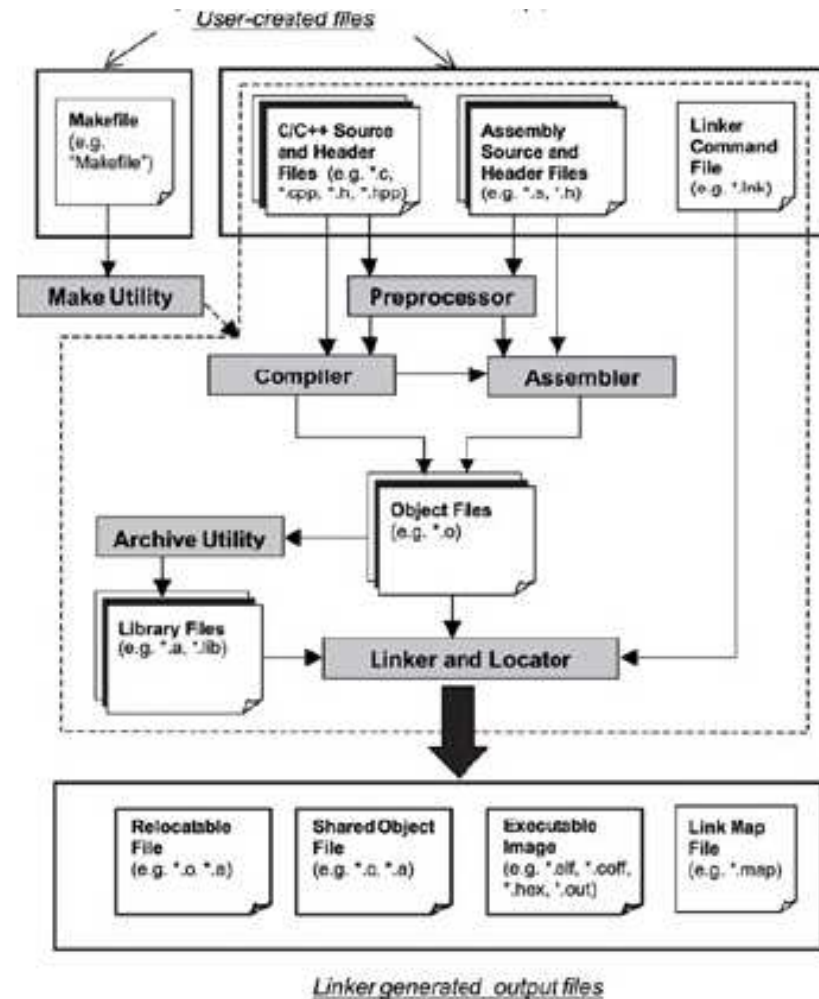
7.2. Reguli de dezvoltare software

7.2.1. Dezvoltarea aplicatiilor software – vedere generala

Tool-urile oferite de sistemul “gazda” sunt: cross compiler, linker, source-level debugger

Sistemul tinta (sistmul embedded) contine: loader dinamic, monitor si un agent pentru debug.

Aplicatiile trebuie mai intai dezvoltate, apoi compilate in cod obiect si apoi linkedate in programe executabile pentru sistemul embedded.

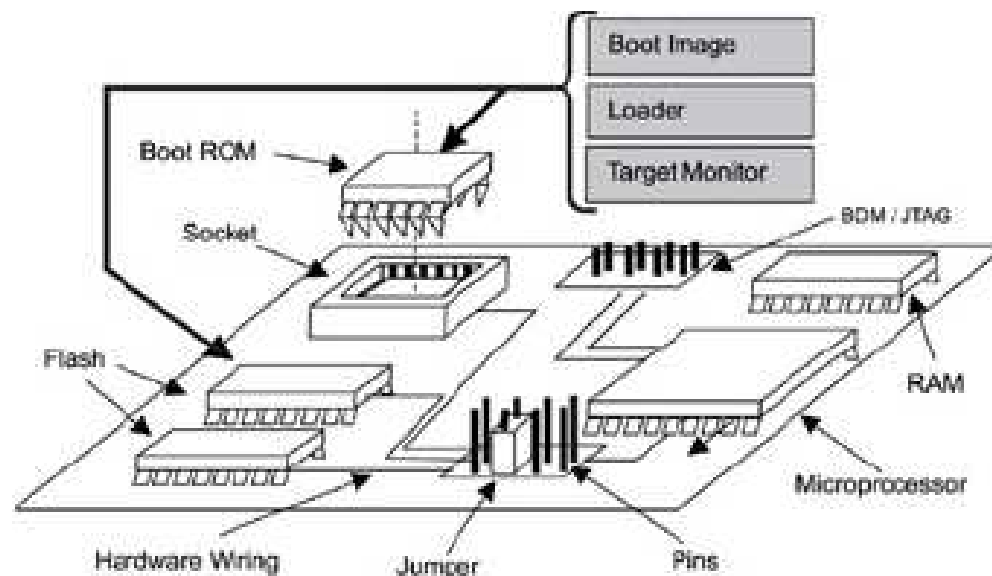


7.2. Reguli de dezvoltare software

7.2.2. Initializarea aplicatiilor software – vedere generala

O imagine executabila construita pentru sistemul emebedded poate fi transferata prin:

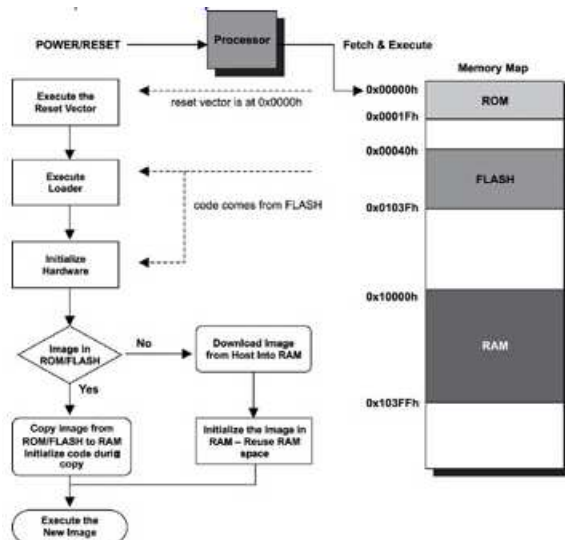
1. Programarea intregii imagini in EEPROM sau fmemoria flash
2. Descarcarea imaginii prin conexiune seriala (RS232), CAN etc.
3. Descarcarea imaginii prin interfete JTAG sau BDM



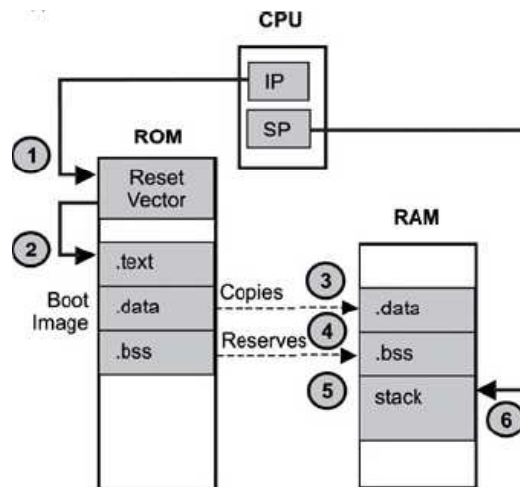
7.2. Reguli de dezvoltare software

7.2.3. Metode de bootare pentru aplicatii software – vedere generala

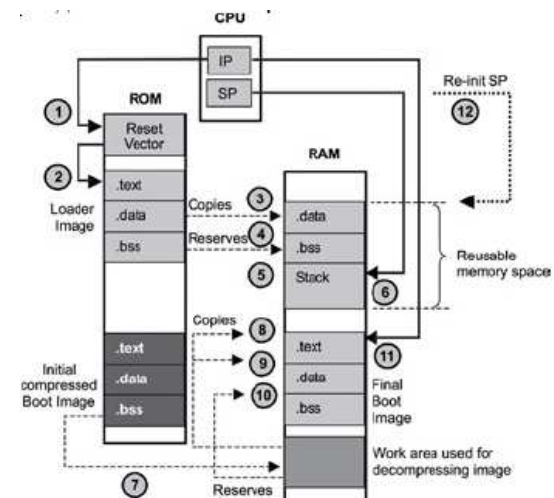
Metoda bootstrap



Rularea codului din ROM
folosind date din RAM



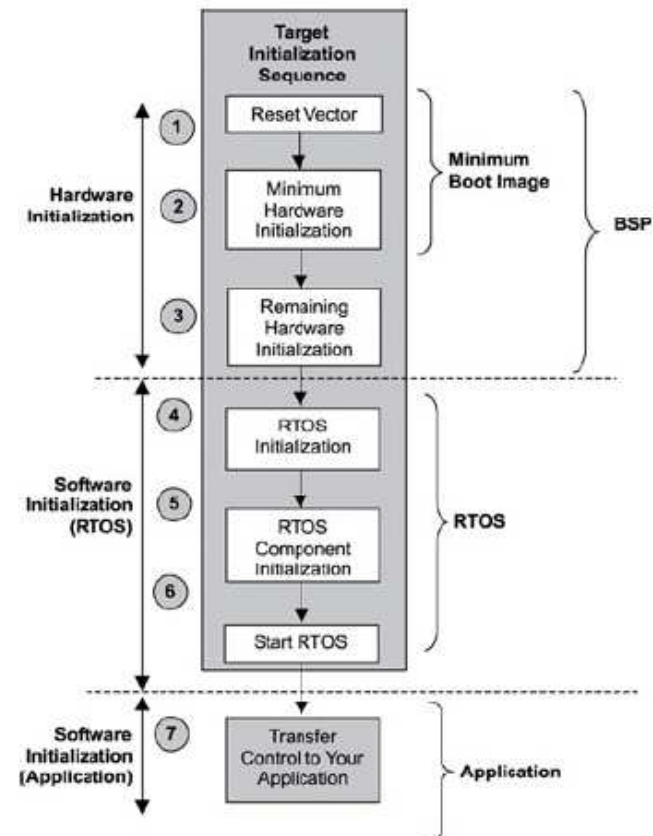
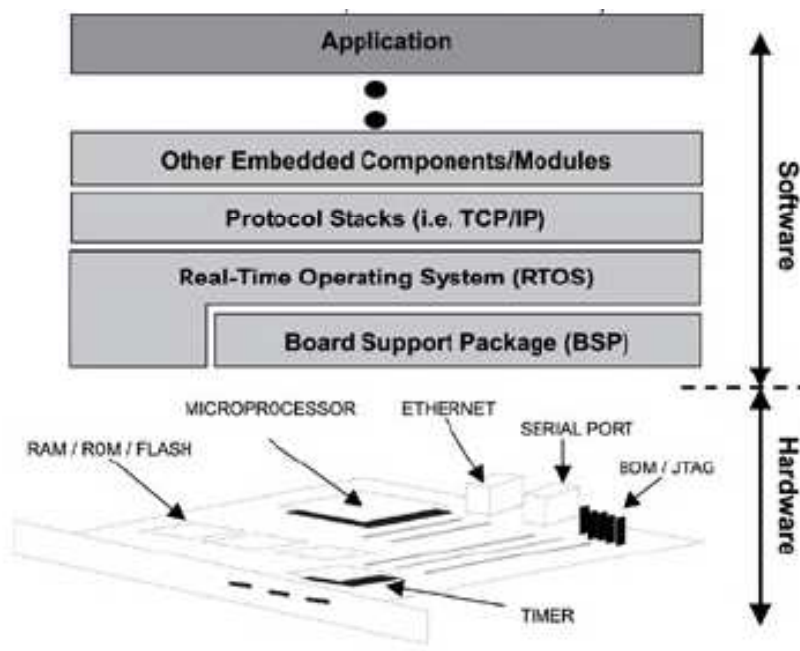
Rularea din RAM dupa
trasferul imaginii din
ROM



7.2. Reguli de dezvoltare software

7.2.4. Initializarea aplicatiilor software – detalii

Principalii pasi in initializarea software-ului sistemelor embedded sunt: initializarea hard, initializarea OS sau RTOS – daca exista si initializarea aplicatiei.



7.2. Reguli de dezvoltare software

7.2.5. Utilizarea mesajelor

Mesajele pot fi transmise/receptionate de catre taskuri sau ISR. Mesajele sunt trimise catre *sending message objects* si receptionate de la *receiving message objects*. Identificarea mesajelor obiect se face folosind identificatori de mesaj, identificatori asignati la generarea sistemului. Catre acelasi *sending message objects* pot fi trasmise mai multe mesaje. Un *receiving message objects* primeste mesaje de la un singure mesaj obiect (cazul comunicatiei interne) sau un singur I-PDU. Un *receiving message objects* pote fi definit ca si *queued*(poate fi citit o singura data dupa care este sters) sau *unqueued*(poate fi citit de mai multe ori).

Un mesaj extern poate avea una din urmatoarele doua proprietati:

- *Triggered Trasfer Property* – mesajul in IPDU-ul corespunzator asignat este updatat si o cerere de trasmisie catre I-PDU este fauta.
- *Pending Trasfer Property* - mesajul in IPDU-il corespunzator este updatat fara o cerere de trasmisie.

Mesajele interne nu au proprietati de transfer. Ele sunt rutate instantaneu catre partea de iesire.

Exista trei moduri de trasmisie petru I-PDU:

- modul de trasmisie direct – trasmisia initializata explicit prin trasmiterea unui mesaj cu proprietatea *Triggered Trasfer Property*
- modul de trasmisie periodic – trasmisie cu o perioada presetata
- modul de trasmisie mixt – combinatie intre primele doua moduri

7.2. Reguli de dezvoltare software

7.2.5. Utilizarea mesajelor

Numai pentru comunicatia externa:

- Receptia incepe cu un indicator de receptionare. Daca acest indicator nu contine o eroare mesajul este copiat in I-PDU.
- Deadline Monitoring poate invoca o eroare de receptionare a mesajului.
- Mesajul este desfacut – conversie din reprezentarea la nivelul retelei la reprezentarea la nivelul CPU

Atat pentru comunicatia interna cat si externa

- Filtrare aplicata continutului mesajelor (in unele cazuri).
- Daca nu se foloseste filtrarea mesajelor, atunci continutul este copiat in mesajul obiect corespunzator
- Daca se foloseste filtrarea, o notificare de receptionare este invocata. Notificarea este per obiect mesaj.
- Datele din mesajul obiect sunt copiate in mesajele aplicatiei.

Filtrarea – inlaturarea anumitor mesaje cand anumite conditii, setate in filtrul de mesaje, nu sunt indeplinite. Filtrarea se face per mesaj obiect. Atributele ce pot fi folosite in implementarea algoritmilor de filtrare sunt:

- Valoarea noua
- Valoarea veche
- Masca,min, max, perioada, offset, etc.
- Numar de receptionari

7.2. Reguli de dezvoltare software

7.2.5. Utilizarea mesajelor

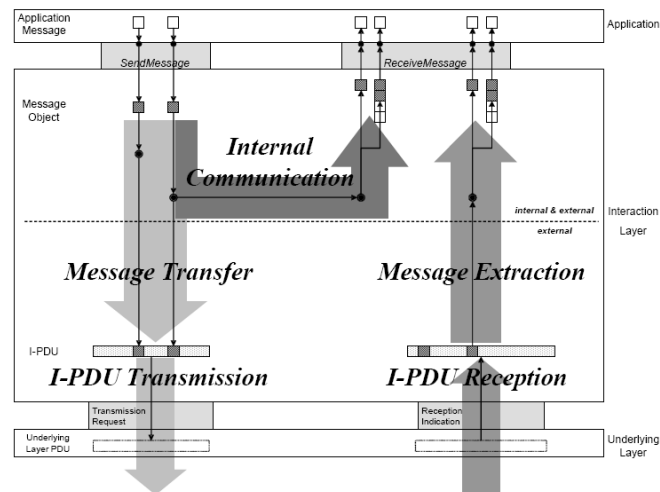
Trasmiterea unui mesaj implica transferul datelor de la nivelul aplicatiei catre I-PDU (pentru comunicatia externa) si/sau *receiving message object(s)* (comunicatie interna).

Numai pentru comunicatia externa:

- Filtare
- CPU order Message Callout
- Conversie intre reprezentari
- Network-order Message Callout – mesajul este memorat in I-PDU

Moduri de trasmsie:

1. Modul direct
2. Modul periodic
3. Modul mixt

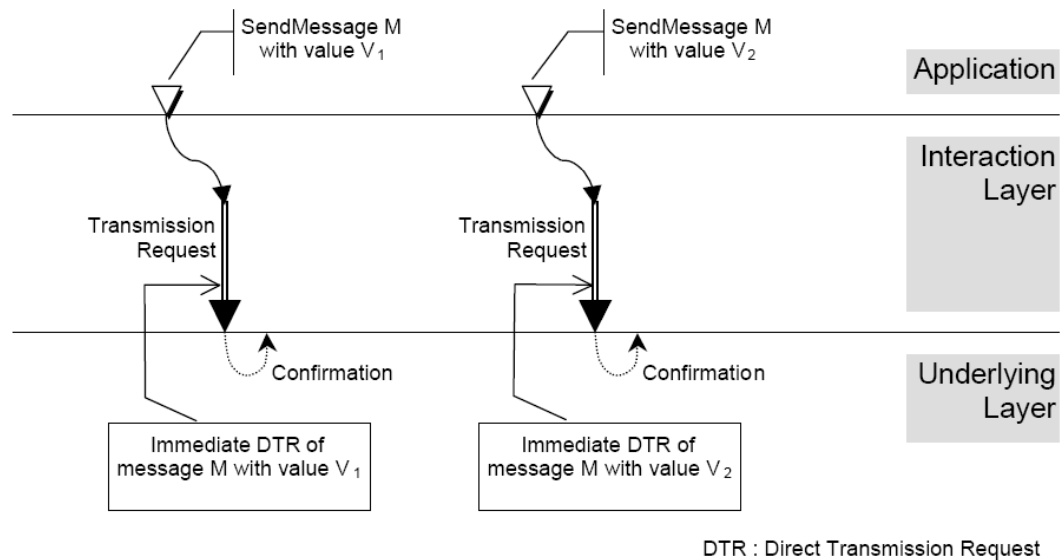
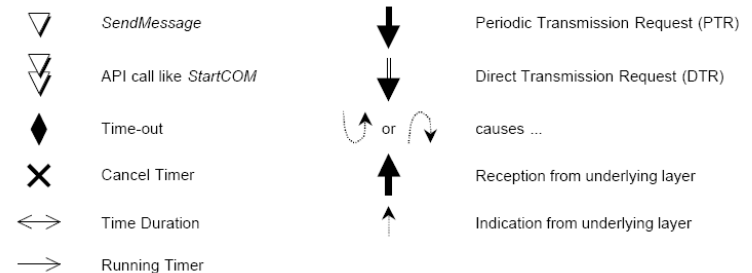


7.2. Reguli de dezvoltare software

7.2.5. Utilizarea mesajelor

Moduri de trasmsie:

1. Modul direct

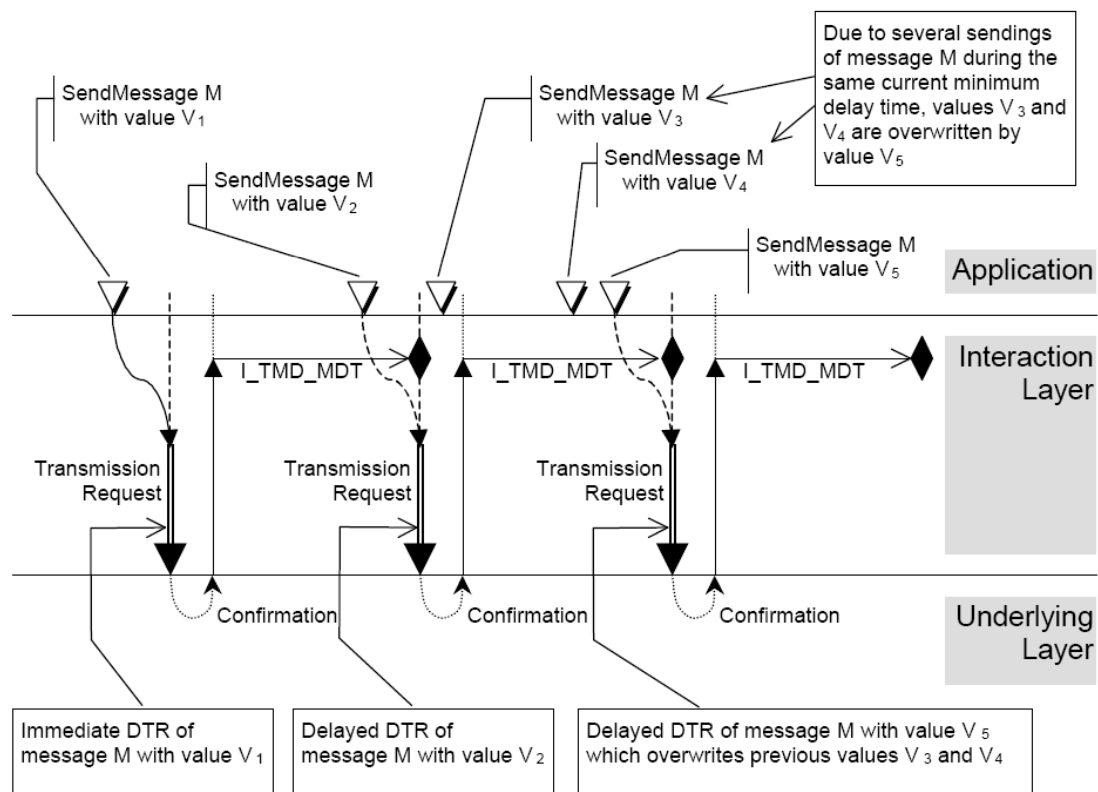
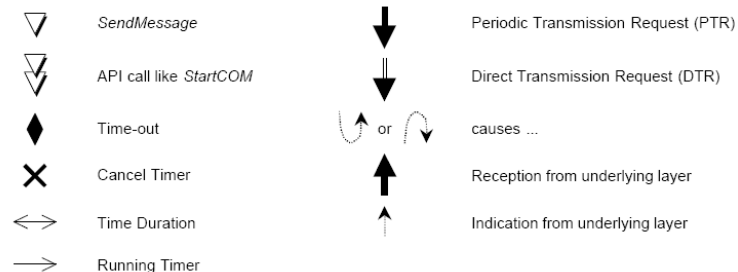


7.2. Reguli de dezvoltare software

7.2.5. Utilizarea mesajelor

Moduri de trasmsie:

1. Modul direct

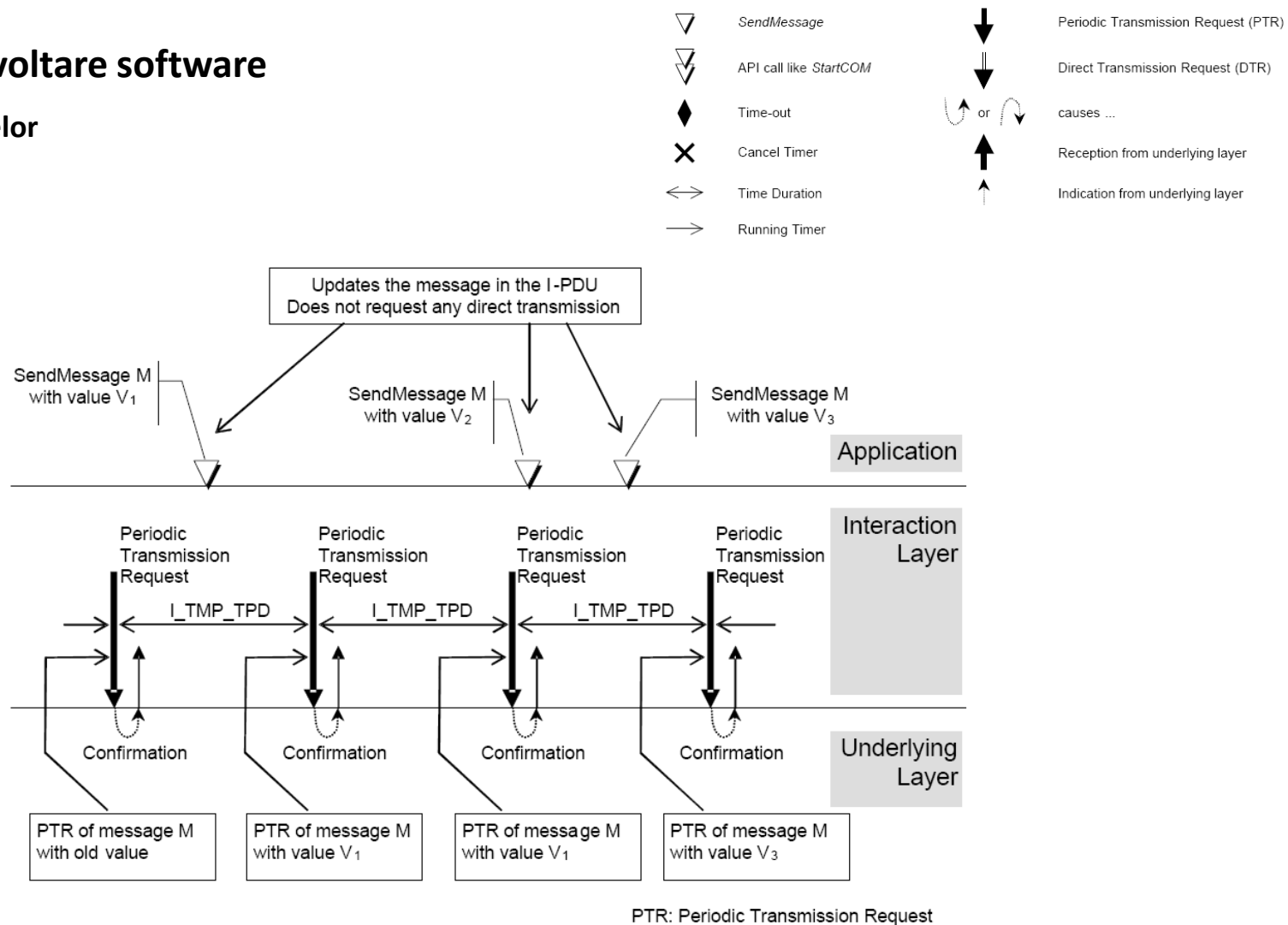


7.2. Reguli de dezvoltare software

7.2.5. Utilizarea mesajelor

Moduri de trasmsie:

2. Modul periodic

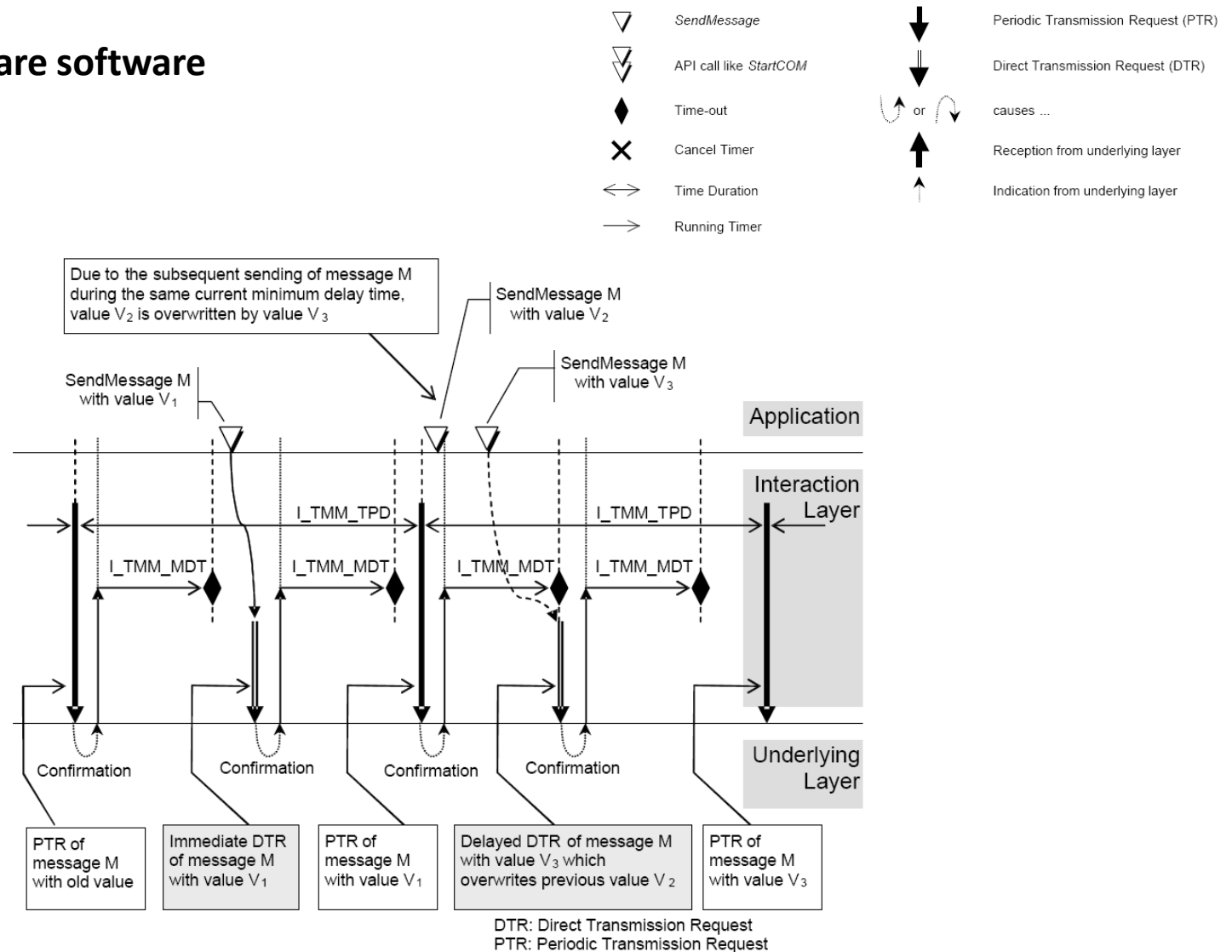


7.2. Reguli de dezvoltare software

7.2.5. Utilizarea mesajelor

Moduri de trasmsie:

3. Modul mixt

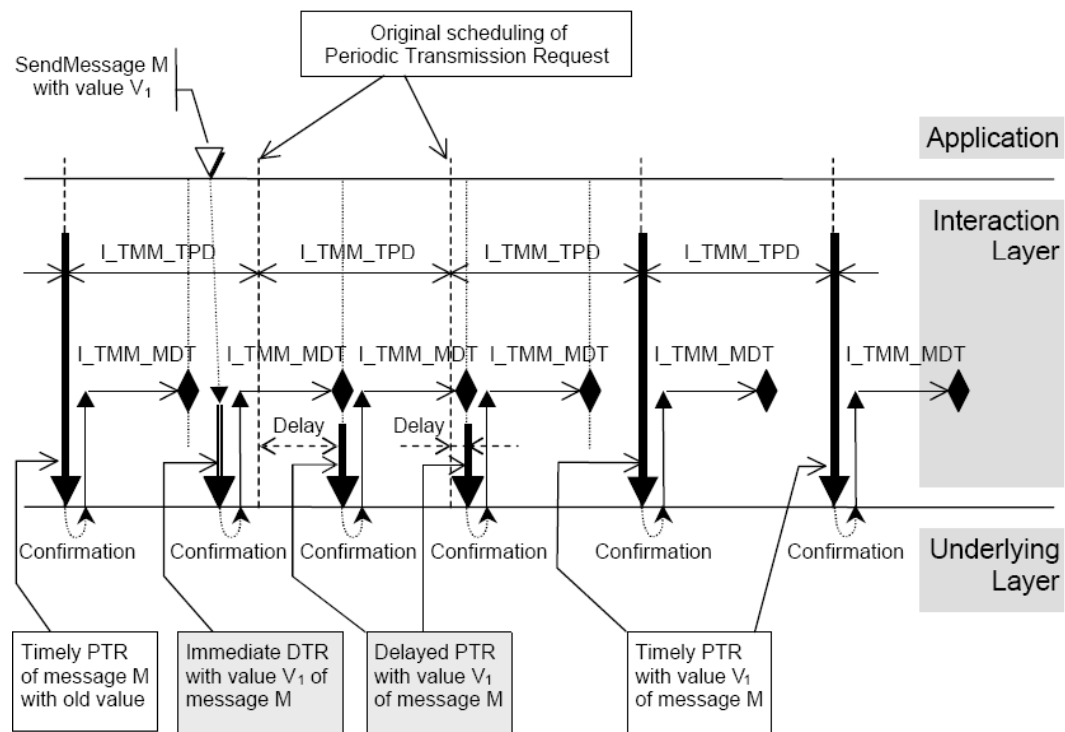
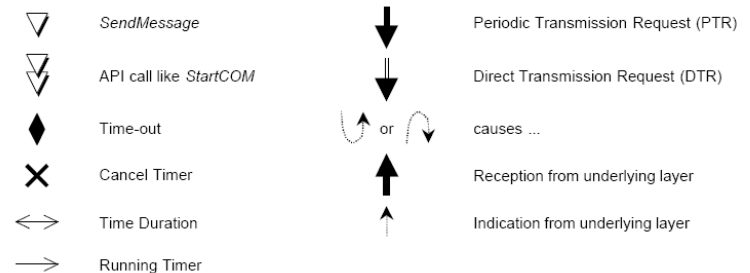


7.2. Reguli de dezvoltare software

7.2.5. Utilizarea mesajelor

Moduri de trasmsie:

3. Modul mixt



DTR: Direct Transmission Request
PTR: Periodic Transmission Request

7.2. Reguli de dezvoltare software

7.2.6. Conversii

Little-endian byte order

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	7	6	5	4	3	2	1	0
Byte 1	15 ← 2	14 ← 1	13 ← 0 LSB	12	11	10	9	8
Byte 2	23 ← 10	22 ← 9	21 ← 8	20 ← 7	19 ← 6	18 ← 5	17 ← 4	16 ← 3
Byte 3	31	30	29	28	27	26	25	24 ← 11 MSB
Byte 4	39	38	37	36	35	34	33	32

Big-endian byte order

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	7	6	5	4	3	2	1	0
Byte 1	15	14	13 ← 11 MSB	12 ← 10	11 ← 9	10 ← 8	9 ← 7	8 ← 6
Byte 2	23 ← 5	22 ← 4	21 ← 3	20 ← 2	19 ← 1	18 ← 0 LSB	17	16
Byte 3	31	30	29	28	27	26	25	24
Byte 4	39	38	37	36	35	34	33	32